Introducción a ML y GenAI

Clasificación - KNN

Ariel Ramos Vela

17-09-2024

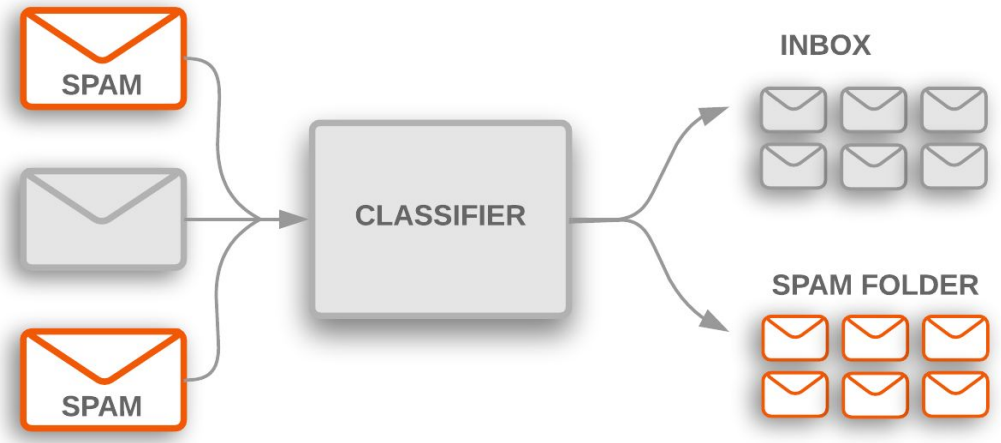
Agenda

1. Introducción a la Clasificación
2. Modelos de Clasificación Comunes
3. Division de datos: Entranamiento, Validación y Prueba.
4. KNN algorithm
5. Evaluación de Modelos de Clasificación
6. Ejemplo: Dataset del Titanic
7. Conclusiones
8. Taller 3

¿Qué es la Clasificación?

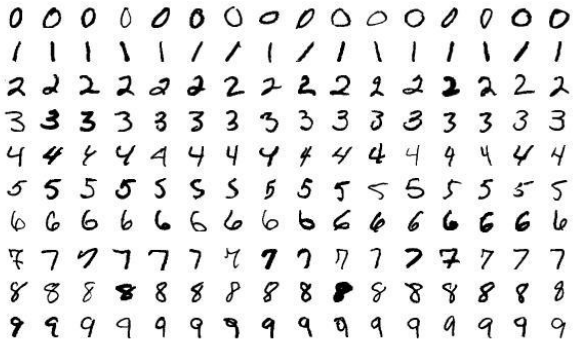
- **Definición:** La clasificación es una técnica de machine learning **supervisada** que asigna una etiqueta (categoría) a una nueva observación basada en datos etiquetados previos.

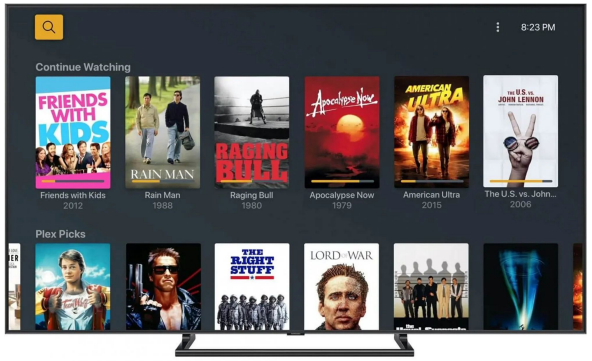
- **Ejemplo cotidiano:** Clasificación de correos electrónicos como spam o no spam.



Tipos de Problemas de Clasificación

- **Binaria:** Solo hay dos clases posibles (0 o 1).
  - Ejemplo: Dataset Titanic
- **Multiclase:** Más de dos clases (p.ej., clasificación de dígitos escritos a mano).
  - Ejemplo: Dataset MNIST
- **Multietiqueta:** Cada instancia puede pertenecer a varias clases al mismo tiempo.
  - **Dataset de Películas (MovieLens):** Clasificar una película en múltiples géneros (acción, comedia, drama).





Pipeline de un Modelo de Clasificación

Recopilación de datos.

Preprocesamiento:

Limpieza de datos

Transformación (p.ej., one-hot encoding)

División de datos:

Conjunto de entrenamiento (80%) / Conjunto de prueba (20%)

Entrenamiento del modelo.

Evaluación del modelo.

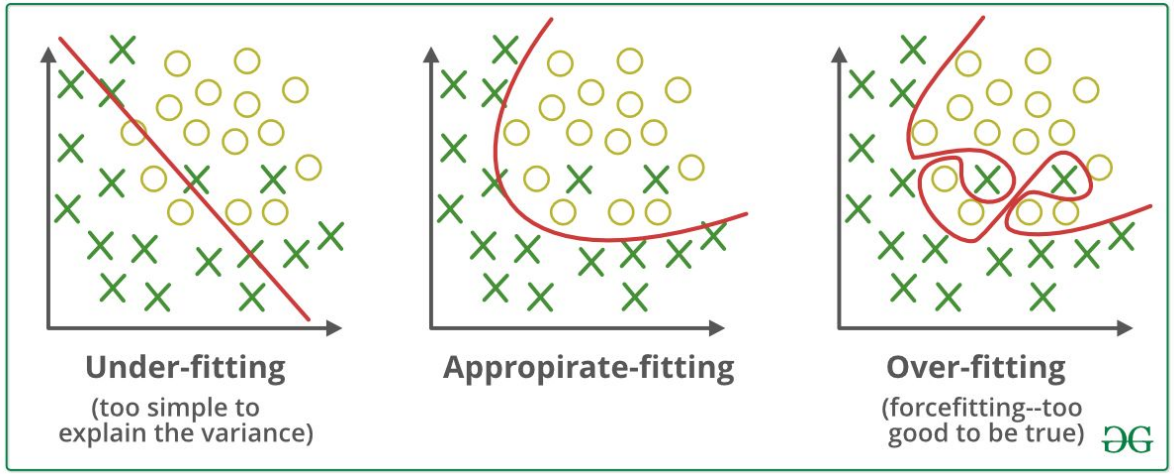
Predicción en datos nuevos.

Modelos de Clasificación Comunes

- **Regresión Logística:** Para problemas de clasificación binaria.
  - Salida: Probabilidades de pertenencia a cada clase.
- **k-Nearest Neighbors (k-NN):** Clasificación basada en la proximidad de los puntos de datos.
- **Árboles de Decisión:** División de los datos en función de características importantes.
- **Random Forest:** Conjunto de múltiples árboles de decisión.
- **Máquinas de Soporte Vectorial (SVM):**Encuentra un hiperplano óptimo que separe las clases.

División de Datos: Entrenamiento, Prueba y Validación

- **¿Por qué dividir los datos?**
- **Evitar el sobreajuste:** Un modelo que se entrena en **todos** los datos puede aprender demasiado bien los detalles específicos del conjunto de datos, haciéndolo menos efectivo para predecir sobre nuevos datos.

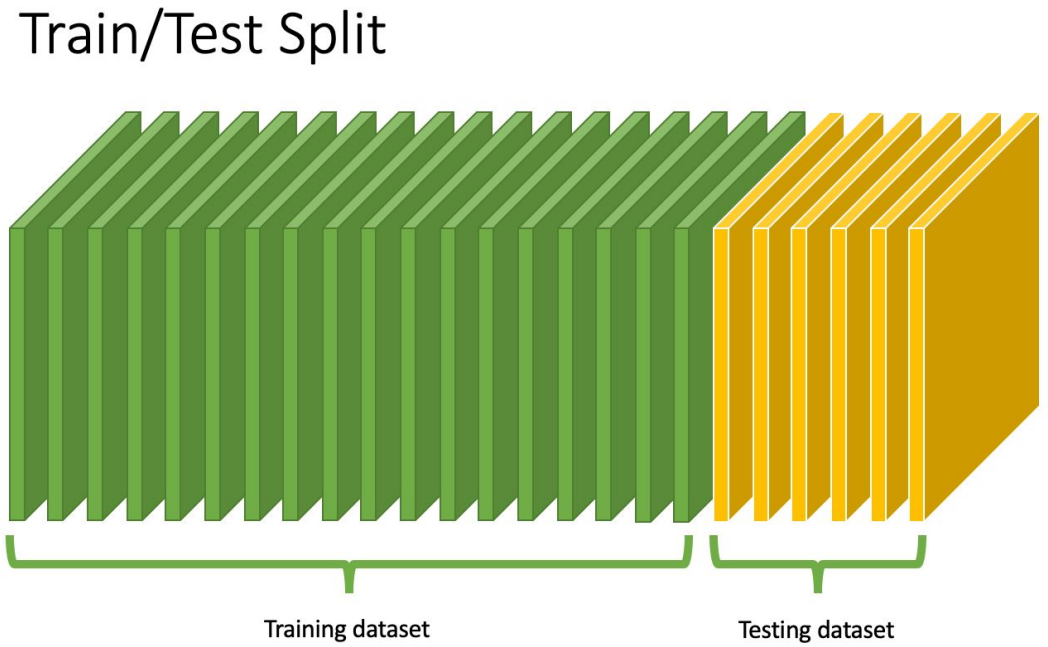


- **Evaluar el rendimiento real:** Necesitamos un conjunto de datos **separado** para evaluar cómo se comporta el modelo en datos que nunca ha visto.

Cómo se hace la división de los datos?

- **Entrenamiento (80%)**: Se utiliza para **entrenar** el modelo, ajustando los parámetros internos.

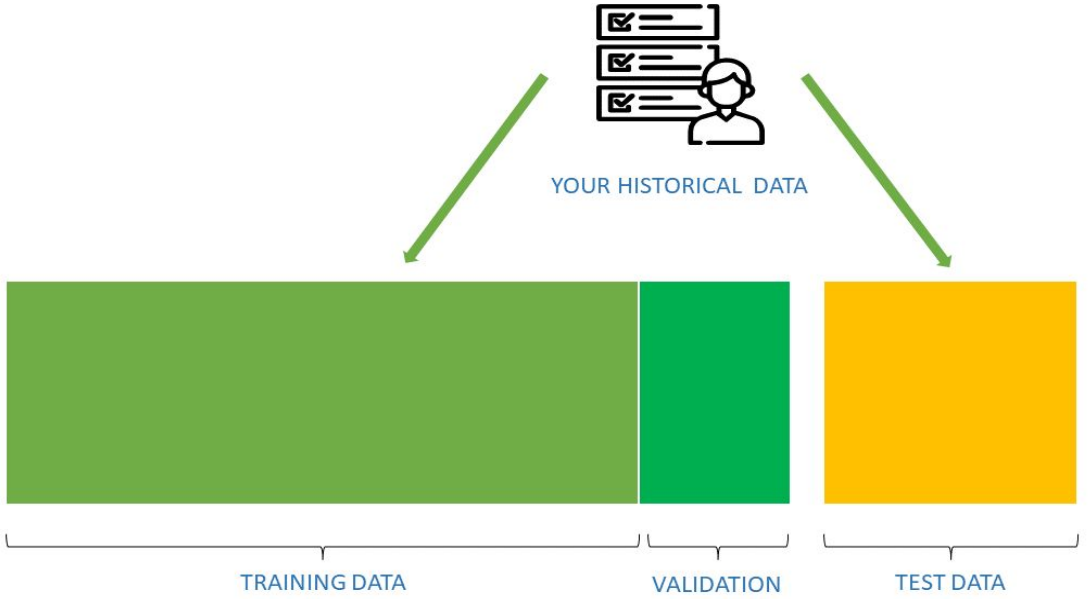
- **Prueba (20%)**: Se usa para **evaluar** el modelo una vez que ha sido entrenado, proporcionando una estimación del rendimiento en datos no vistos.



¿Qué es un conjunto de Validación?

- **Validación (10-20%)**: A veces, se añade un conjunto adicional de datos llamado **conjunto de validación**. Este se usa durante el entrenamiento para ajustar hiperparámetros (como el valor de k en k-NN) sin afectar el conjunto de prueba.

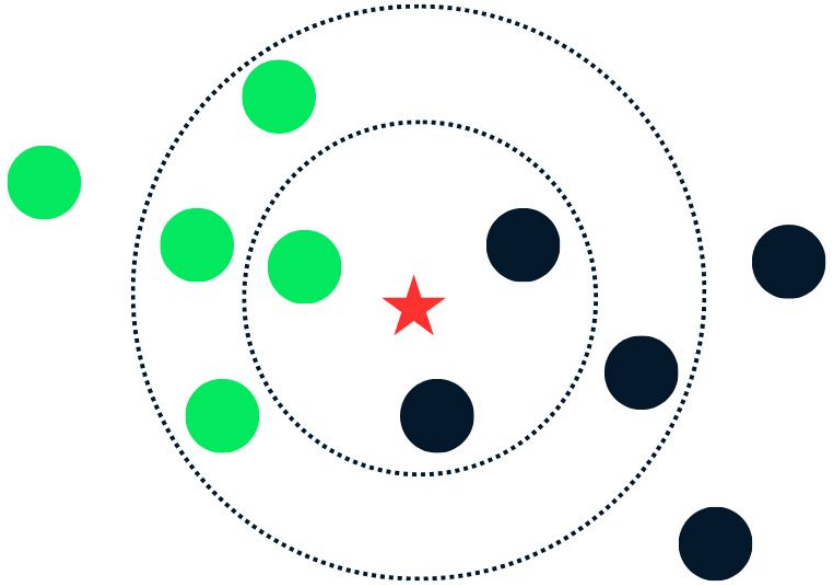
- **División típica:**
  - **Entrenamiento (60-70%)**
  - **Validación (10-20%)**
  - **Prueba (20%)**



Introducción a k-Nearest Neighbors (k-NN)

- **k-Nearest Neighbors (k-NN)** es un **algoritmo de aprendizaje supervisado** utilizado tanto para **clasificación** como para **regresión**.

- **Principio básico:** Dado un punto nuevo, el algoritmo busca los **k puntos más cercanos** en el conjunto de datos de entrenamiento y toma una decisión basada en la mayoría (para clasificación) o el promedio (para regresión) de sus etiquetas.



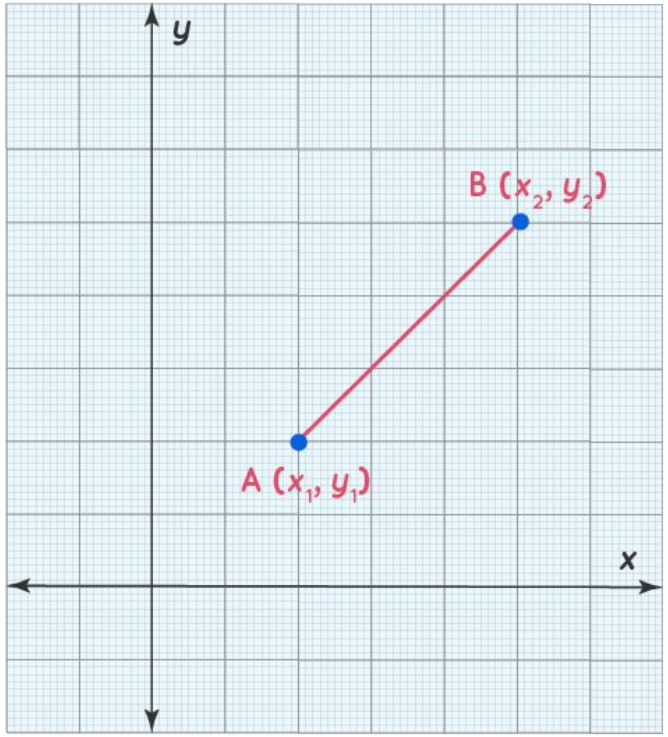
Funcionamiento de k-NN

1. **Recoge datos etiquetados** (conjunto de entrenamiento).
2. **Elige un valor de k** (el número de vecinos más cercanos).
3. **Calcula la distancia** entre el nuevo punto y cada punto del conjunto de datos.
4. **Selecciona los k vecinos más cercanos** (usualmente con la distancia Euclidiana).
5. **Clasificación o predicción:**
    1. **Clasificación:** Elige la clase más común entre los k vecinos.
    2. **Regresión:** Promedia los valores de los k vecinos.

Cálculo de la Distancia (Distancia Euclidiana)

- **Distancia Euclidiana** es la más utilizada para medir la cercanía entre puntos.

$$d(A, B) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$



Ejemplo:

Altura	Peso	Clase
160 cm	55 kg	Deportista
170 cm	70 kg	No deportista
165 cm	60 kg	?

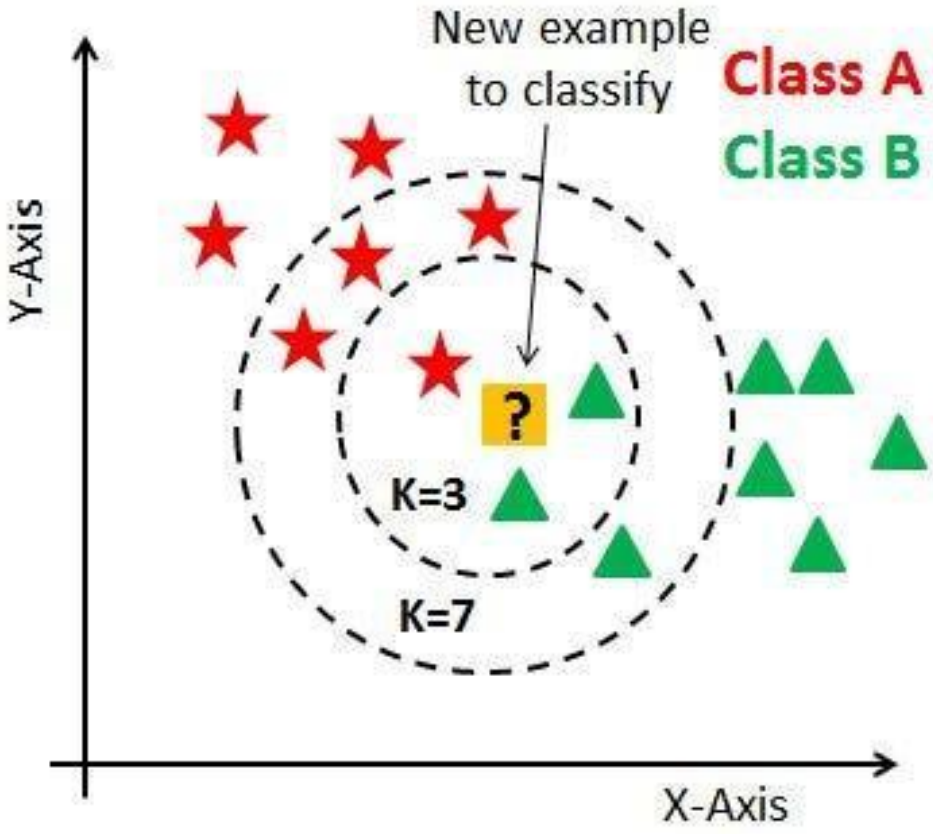
- Para este ejemplo, elegimos **k = 1**, lo que significa que clasificaremos la nueva muestra en la clase de su vecino más cercano.

- ¿Cuál es el vecino más cercano?

Elección del valor de k

- **Elección de k:** Es crucial elegir el valor de **k** correcto, ya que afecta el rendimiento del modelo.
- **k pequeño (p.ej., k=1):** El modelo puede sobreajustarse (problema con outliers)
- **k grande:** El modelo puede volverse demasiado general.

- ¿Hay algún problema si **k** es un número par?
- ¿Qué pasa si **k** es igual al número de datos que tenemos?



Pros y Contras de k-NN

- **Ventajas:** Sencillo de implementar.
  - No hace ninguna suposición sobre la distribución de los datos.
  - Funciona bien con datos pequeños y bien distribuidos.

- **Desventajas:**
  - **Lento en tiempo de predicción:** k-NN necesita calcular distancias para todos los puntos en el conjunto de datos.
  - Sensible a la **escala de las características** (se recomienda normalizar los datos).
  - Requiere una buena elección de **k**.

Evaluación de Modelos de Clasificación

1. **Exactitud (Accuracy)**
   - **Descripción:** Mide la proporción de predicciones correctas entre todas las predicciones realizadas.

$$\text{Exactitud} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}$$

   - **Interpretación:** Ideal para datasets balanceados. Sin embargo, no es adecuada si las clases están desbalanceadas.

TP: True positive

TN: True negative

FP: False positive

FN: False negative

2. Precisión (Precision)

- **Descripción:** Mide la proporción de predicciones positivas correctas sobre el total de predicciones positivas.

- **Interpretación:** Alta precisión significa que hay pocos **falsos positivos**. Es útil cuando los **falsos positivos** son costosos, como en un diagnóstico médico.

$$\text{Precisión} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

3. Exhaustividad (Recall)

- **Descripción:** Mide la proporción de verdaderos positivos capturados sobre el total de verdaderos positivos.

- **Interpretación:** Alta exhaustividad significa que el modelo captura la mayoría de los **verdaderos positivos**. Es importante cuando los **falsos negativos** son costosos, como en la detección de fraudes.

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

4. F1-Score

- **Descripción:** Es el promedio armónico entre la precisión y el recall. Proporciona un balance entre ambas métricas.

- **Interpretación:** El F1-Score es útil cuando se busca un equilibrio entre la precisión y el recall, especialmente en datasets desbalanceados.

$$\text{F1-Score} = 2 \times \frac{\text{Precisión} \times \text{Recall}}{\text{Precisión} + \text{Recall}}$$

5. Matriz de Confusión

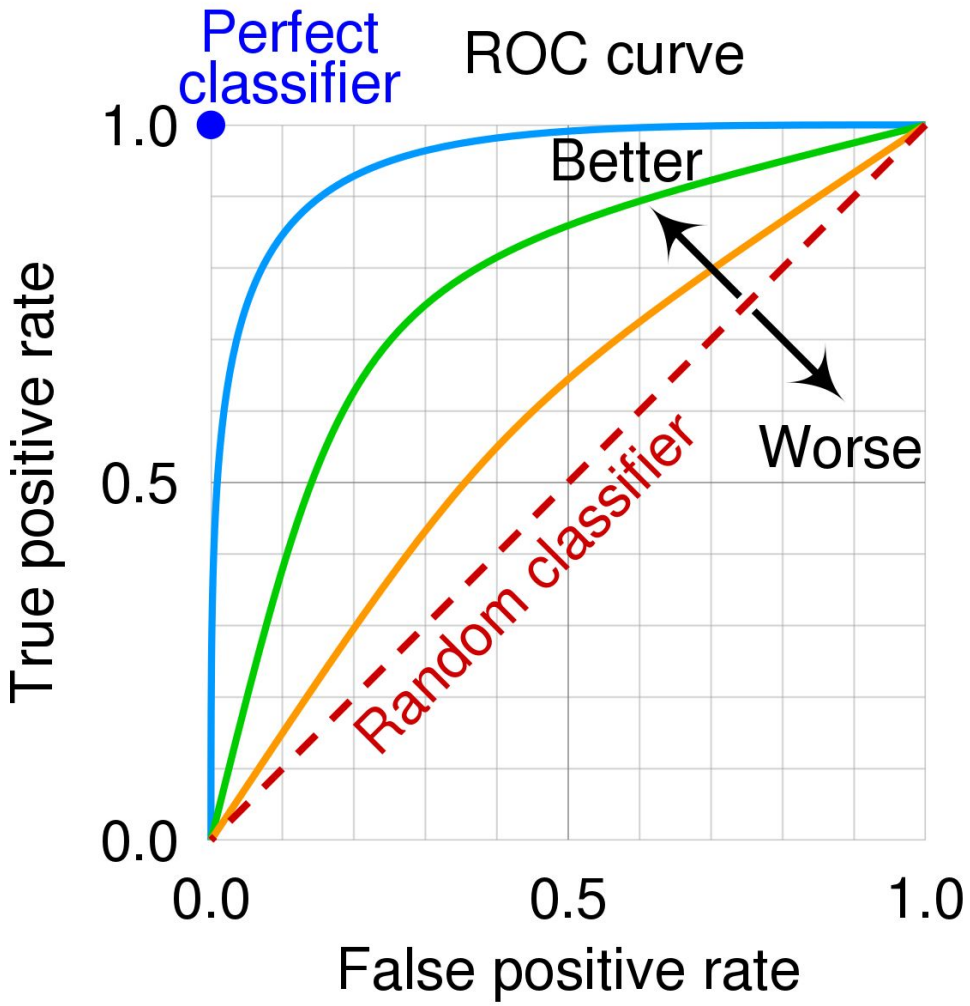
- **Descripción:** Es una tabla que muestra el rendimiento del modelo al clasificar instancias en clases reales y predichas.
- **Interpretación:** Permite identificar cuántas predicciones fueron correctas o incorrectas para cada clase, lo que facilita ver errores de clasificación y sesgos del modelo.

	Predicción: Sobrevive	**Predicción: No sobrevive**
Real: Sobrevive	35 (TP)	15 (FN)
Real: No sobrevive	10 (FP)	50 (TN)

6. AUC-ROC (Área bajo la curva - Receiver Operating Characteristic)

- **Descripción:** Mide la capacidad del modelo para distinguir entre clases. ROC es una curva que muestra la relación entre **TPR (Tasa de verdaderos positivos)** y **FPR (Tasa de falsos positivos)**.

- **Interpretación:** Un valor de AUC cercano a 1 indica un modelo excelente, mientras que un valor de 0.5 indica un modelo que no tiene mejor desempeño que el azar.



Entrenamiento del Modelo con el Dataset Titanic

- **Paso 1:** División en conjunto de entrenamiento (80%) y prueba (20%).
- **Paso 2:** Selección del modelo (KNN).

- **Paso 3:** Entrenamiento del modelo (**Usando los datos de entranamiento**).

- **Paso 4:** Predicción en el conjunto de prueba.

- **Paso 5:** Evaluación del modelo.
  - Interpreta las métricas.

- **Paso 6:** Interpretación del modelo
  - **Discusión:** Cómo se interpretan estos resultados en el contexto del Titanic? (Con relación a estos features: Sex, Pclass, Age)

Conclusiones

- La **clasificación** es una técnica fundamental en Machine Learning que tiene aplicaciones en muchos campos.
- **Modelos** como la Regresión Logística, k-NN, Árboles de Decisión y SVM son herramientas útiles dependiendo del problema.
- La **evaluación** del modelo es esencial para entender su rendimiento y hacer mejoras.